

First name and last name

Fill out with capital letters

Student number

Writing time: 75 minutes, date: January 6, 2019

Comments: in case of all programs assume that libraries `iostrem` and `stdlib` are attached and the namespace `std` is available. Only places marked for the answers are graded. In case you find a mistake or ambiguity in a question, please write an appropriate comment which explains that. The number of marks for the test is 0-100 points (passing threshold = 50%).

Question 1. (20 pts. = 7*2 + 6)

Selected fields fill out so that the program compiles correctly and prints out to the screen:

AB

When giving your answer write **ERROR** in all fields if there is no solution.

Comment: each field has score of 2 pts. Correct output to the screen gives 6 pts.

Comment: we assume that each call to *malloc* succeeds. Note that the program does not free the memory, which does not affect its analysis here (we skip releasing memory due to limited space).

```
typedef struct A {
    char c;
    struct A *next;
} list_t;
list_t *add( list_t *h, _____ ) {
    list_t *n = (list_t *)malloc( sizeof(list_t) );

    _____ = _____;
    n->c = c;
    return n;
}
int main() {
    list_t *l = NULL;

    _____ = add( _____, 'A' );
    _____ = add( _____, 'B' );
    while ( l != NULL ) {
        cout << l->c;
        l = l->next;
    }
    return 0;}
```

Question 2. (20 pts. = 4*5 pts.)

What are the contents of the array *a* right before the end of the execution of function *main*.

Answer:

a[0] = _____

a[1] = _____

a[2] = _____

a[3] = _____

```
void m( int *a, int *b, int *c, int n ){
    int i;
    for (i=0; i < n; i++ ) {
        if ( b[i] % 3 > c[i] % 3 )
            a[i] = b[i];
        else
            a[i] += c[i];
    }
}
int main() {
    int a[] = { 1,2,0,2, 2,1,0,1, 2,0,2,1 };
    m( a+8, a+4, a, 4 );
    m( a+4, a+8, a, 4 );
    m( a, a+4, a+8, 4 );
    return 0;
}
```

Question 3. (20 pts. = 2*10 pts.)

Determine what will be written to the screen in the particular calls to *cout*.

First call to
cout:

Second call to
cout:

```
int count( int *a, int n ) {
    if ( n > 1 )
        return count( a+1, n-1 ) + ( a[0]>a[1] ? 1:0 );
    else
        return 0;
}
int main() {
    int x[] = { 3,5,8,4,6,3,3,9,6,3,7,6,1 };
    cout << count( x, (sizeof x) / (sizeof x[1]) );
    cout << count( &(x[2]), 7 );
    return 0;
}
```

Question 4. (20 pts. = 4*5 pts.)

Selected fields fill out in such a way that the program compiles correctly, executes without errors and when executed writes to the screen the number 7. When giving your answer:

- do not use: ;,[]
- write *ERROR* if no solution exists
- write *EMPTY* if a field should remain empty.

```
int g( _____ ) {
    _____;
}
int f( _____ ) {
    g( _____ );
    return x;
}
int main() {
    cout << f( 5 );
    return 0;
}
```

Question 5. (20 pts. = 5*4 pts.)

Give the text which is written to the screen as a result of executing subsequent instructions *cout* (in place for an answer marked with label "Instruction x" write the text that is printed to the screen by instruction *cout* marked with comment */* I-x */*. Write *ERR* if the answer cannot be uniquely determined. Binary encoding of numbers is assumed to be as presented during the lectures, i.e., U2. If some instruction results in writing outside an array or some other execution error, then also write *ERR* as an answer and continue your analysis by skipping this instruction.

```
#define W u+h
int main() {
    int *p = (int *) malloc( 100*sizeof(int) );
    int h=5, u=4, *v = &(p[30]), y[] = {4,3,2,1,0};

    cout << (12 ^ 3); /* I-1 */
    v += (h << 2);
    cout << v; /* I-2 */
    cout << (v-p)/sizeof(int); /* I-3 */
    cout << ( W*W ); /* I-4 */
    cout << &(y[3]) - &(*y); /* I-5 */
    return 0;
}
```

Answers:

Instruction 1: _____

Instruction 2: _____

Instruction 3: _____

Instruction 4: _____

Instruction 5: _____